



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

+partition +(code restructuring) +edge +node +weight



THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [partition](#) [code restructuring](#) [edge](#) [node](#) [weight](#)

Found 842 of 141,345

Sort results
by

relevance

Display
results

expanded form



[Save results to a Binder](#)



[Search Tips](#)

☐ Open results in a new
window

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Complete removal of redundant expressions](#)

Rastislav Bodík, Rajiv Gupta, Mary Lou Soffa

May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference
on Programming language design and implementation**, Volume 33 Issue 5

Full text available: pdf(2.13 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Partial redundancy elimination (PRE), the most important component of global optimizers, generalizes the removal of common subexpressions and loop-invariant computations. Because existing PRE implementations are based on *code motion*, they fail to completely remove the redundancies. In fact, we observed that 73% of loop-invariant statements cannot be eliminated from loops by code motion alone. In dynamic terms, traditional PRE eliminates only half of redundancies that are strictly partial. ...

Keywords: control flow restructuring, demand-driven frequency data-flow analysis, partial redundancy elimination, profile-guided optimization, speculative execution

2 [Global optimizations for parallelism and locality on scalable parallel machines](#)

Jennifer M. Anderson, Monica S. Lam

June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference
on Programming language design and implementation**, Volume 28 Issue 6

Full text available: pdf(1.74 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

3 [An affine partitioning algorithm to maximize parallelism and minimize communication](#)

Amy W. Lim, Gerald I. Cheong, Monica S. Lam

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.33 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

4 [Supporting the restructuring of data abstractions through manipulation of a program
visualization](#)

Robert W. Bowdidge, William G. Griswold

April 1998 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 7 Issue 2

Full text available:  [pdf\(1.57 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

With a meaning-preserving restructuring tool, a software engineer can change a program's structure to ease future modifications. However, deciding how to restructure the program requires a global understanding of the program's structure, which cannot be derived easily by directly inspecting the source code. We describe a manipulable program visualization—the star diagram—that supports the restructuring task of encapsulating a global data structure. The star diag ...

Keywords: meaning-preserving restructuring, semi-automated restructuring, software visualization, star diagram, tool-supported restructuring

5 [Compiler-directed selection of dynamic memory layouts](#)

Mahmut Kandemir, Ismail Kadayif

April 2001 **Proceedings of the ninth international symposium on Hardware/software codesign**

Full text available:  [pdf\(650.29 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Compiler technology is becoming a key component in the design of embedded systems, mostly due to increasing participation of software in the design process. Meeting system-level objectives usually requires flexible and retargetable compiler optimizations that can be ported across a wide variety of architectures. In particular, source-level compiler optimizations aiming at increasing locality of data accesses are expected to improve the quality of the generated code. Previous compiler-based ap ...

Keywords: array reuse, data dependence, data locality, memory layout optimization, software compilation

6 [Interprocedural transformations for parallel code generation](#)

Mary W. Hall, Ken Kennedy, Kathryn S. McKinley

August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**


Full text available:  [pdf\(1.26 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

7 [Load-reuse analysis: design and evaluation](#)

Rastislav Bodík, Rajiv Gupta, Mary Lou Soffa

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5

Full text available:  [pdf\(1.86 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Load-reuse analysis finds instructions that repeatedly access the same memory location. This location can be promoted to a register, eliminating redundant loads by reusing the results of prior memory accesses. This paper develops a load-reuse analysis and designs a method for evaluating its precision. In designing the analysis, we aspire for *completeness*—the goal of exposing all reuse that can be harvested by a subsequent program transformation. For register promotion, a suitable transfo ...

Keywords: data-flow analysis, profile-guided optimizations, program representations, register promotion

8 [Cross references are features](#)

9 Reducing branch costs via branch alignment

Brad Calder, Dirk Grunwald

November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 ,
28 Issue 11 , 5

Full text available:  pdf(1.20 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Several researchers have proposed algorithms for basic block reordering. We call these branch alignment algorithms. The primary emphasis of these algorithms has been on improving instruction cache locality, and the few studies concerned with branch prediction reported small or minimal improvements. As wide-issue architectures become increasingly popular the importance of reducing branch costs will increase, and branch alignment is one mechanism which can effectively reduce ...

Keywords: branch prediction, branch target buffers, profile-based optimization, trace scheduling

10 A software engineering perspective on algorithmics

Karsten Weihe

March 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 1

Full text available:  pdf(1.62 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)


An algorithm component is an implementation of an algorithm which is not intended to be a stand-alone module, but to perform a specific task within a large software package or even within several distinct software packages. Therefore, the design of algorithm components must also incorporate software-engineering aspects. A key design goal is adaptability. This goal is important for maintenance throughout a project, prototypical development, and reuse in new, unforeseen context ...

Keywords: algorithm engineering

11 Reverse engineering: Search based reverse engineering

Brian S. Mitchell, Spiros Mancoridis, Martin Traverso

July 2002 **Proceedings of the 14th international conference on Software engineering and knowledge engineering**

Full text available:  pdf(208.64 KB)

Additional Information: [full citation](#), [abstract](#), [references](#)

In this paper we describe a two step process for reverse engineering the software architecture of a system directly from its source code. The first step involves clustering the modules from the source code into abstract structures called subsystems. The second step involves reverse engineering the subsystem-level relations using a formal (and visual) architectural constraint language. We use search techniques to accomplish both of these steps, and have implemented a suite of integrated tools to ...


12

Combinational logic synthesis for LUT based field programmable gate arrays

Jason Cong, Yuzheng Ding

April 1996 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**,

Volume 1 Issue 2

Full text available:  pdf(628.91 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The increasing popularity of the field programmable gate-array (FPGA) technology has generated a great deal of interest in the algorithmic study and tool development for FPGA-specific design automation problems. The most widely used FPGAs are LUT based FPGAs, in which the basic logic element is a K-input one-output lookup-table (LUT) that can implement any Boolean function of up to K variables. This unique feature of the LUT has brought new challenges to lo ...

Keywords: FPGA, area minimization, computer-aided design of VLSI, decomposition, delay minimization, delay modeling, logic optimization, power minimization, programmable logic, routing, simplification, synthesis, system design, technology mapping

13 SilkRoute: A framework for publishing relational data in XML

Mary Fernández, Yana Kadiyska, Dan Suciu, Atsuyuki Morishima, Wang-Chiew Tan

December 2002 **ACM Transactions on Database Systems (TODS)**, Volume 27 Issue 4

Full text available:  pdf(687.91 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

XML is the "lingua franca" for data exchange between interenterprise applications. In this work, we describe SilkRoute, a framework for publishing relational data in XML. In SilkRoute, relational data is published in three steps: the relational tables are presented to the database administrator in a canonical XML view; the database administrator defines in the XQuery query language a public, virtual XML view over the canonical XML view; and an application formulates an XQuery query over the publ ...

Keywords: XML, XML storage systems, XQuery

14 All things UML: Nice class diagrams admit good design?

Holger Eichelberger

June 2003 **Proceedings of the 2003 ACM symposium on Software visualization**

Full text available:  pdf(413.61 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Analysis and design of programs by using tools has emerged to a standard technique in object-oriented software engineering. Many of these tools claim to implement methods according to the UML standard and some of the tools provide automatic layout of the diagrams drawn by the user or generated automatically from source code. In this paper we propose a set of aesthetic criteria for UML class diagrams and discuss the relation between these criteria, HCI and design aspects of object-oriented softwa ...

Keywords: HCI, UML class diagrams, aesthetics, metrics, software engineering

15 Cache-conscious data placement

Brad Calder, Chandra Krintz, Simmi John, Todd Austin

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32

Issue 11 , 5

Full text available:  pdf(1.49 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As the gap between memory and processor speeds continues to widen, cache efficiency is an

increasingly important component of processor performance. Compiler techniques have been used to improve instruction cache performance by mapping code with temporal locality to different cache blocks in the virtual address space eliminating cache conflicts. These code placement techniques can be applied directly to the problem of placing data for improved data cache performance. In this paper we present a gene ...

16 Technical reports

SIGACT News Staff

January 1980 **ACM SIGACT News**, Volume 12 Issue 1

Full text available:  pdf(5.28 MB) Additional Information: [full citation](#)

17 A dynamic locality optimization algorithm for linear algebra codes

Mahmut Kandemir

March 2001 **Proceedings of the 2001 ACM symposium on Applied computing**

Full text available:  pdf(107.74 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: cache locality, data reuse, dynamic optimizations, memory layouts, optimizing compilers

18 Efficient scheduling of conditional behaviors for high-level synthesis

Apostolos A. Kountouris, Christophe Wolinski

July 2002 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 7 Issue 3

Full text available:  pdf(1.50 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As hardware designs get increasingly complex and time-to-market constraints get tighter there is strong motivation for high-level synthesis (HLS). HLS must efficiently handle both dataflow-dominated and controlflow-dominated designs as well as designs of a mixed nature. In the past efficient tools for the former type have been developed but so far HLS of conditional behaviors lags behind. To bridge this gap an efficient scheduling heuristic for conditional behaviors is presented. Our heuristic a ...

Keywords: Design automation, conditional behavior, high level synthesis (HLS), scheduling

19 A compiler technique for improving whole-program locality

Mahmut Taylan Kandemir

January 2001 **ACM SIGPLAN Notices , Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 36 Issue 3

Full text available:  pdf(1.37 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Exploiting spatial and temporal locality is essential for obtaining high performance on modern computers. Writing programs that exhibit high locality of reference is difficult and error-prone. Compiler researchers have developed loop transformations that allow the conversion of programs to exploit locality. Recently, transformations that change the memory layouts of multi-dimensional arrays---called data transformations---have been proposed. Unfortunately, both data and loop transformations have ...

Keywords: cache locality, data reuse, memory layouts, optimizing compilers, static optimizations

20 NanoFabrics: spatial computing using molecular electronics

Seth Copen Goldstein, Mihai Budiu

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available:  pdf(996.26 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)



[Publisher Site](#)

[terms](#)

The continuation of the remarkable exponential increases in processing power over the recent past faces imminent challenges due in part to the physics of deep-submicron CMOS devices and the costs of both chip masks and future fabrication plants. A promising solution to these problems is offered by an alternative to CMOS-based computing, chemically assembled electronic nanotechnology (CAEN).

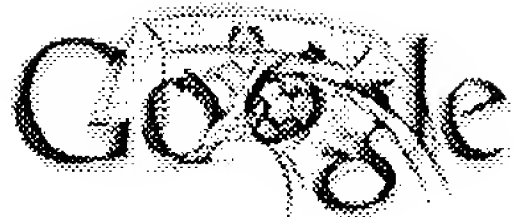
In this paper we outline how CAEN-based computing can become a reality. We briefly describe rec ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Web Images Groups News Froogle more »

partition node edge weight

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 24,300 for **partition node edge weight**. (1.06 seconds)

[PPT] [Exploring Variable Partitioning for Dual Data-Memory Bank ...](#)

File Format: Microsoft Powerpoint 97 - [View as HTML](#)

... NP-Complete problem – Max Cut Problem. Always choose to **partition** the **node** that create the largest total **edge weight** gains between the partitions. 8/16/04.

www.utdallas.edu/~qfzhuge/talks/msp01_present.ppt - [Similar pages](#)

[Ali Dasdan's Partitioning Center](#)

... ll is for the initial **partition**, and FF is for the final **partition**. ... number of **nodes** 'number of **edges**' **edge weight** 'edge degree' '1st **node**' '2nd **node** ...

www.cecs.uci.edu/~dasdan/partition.html - 8k - [Cached](#) - [Similar pages](#)

[PDF] [A Parallel Algorithm for Answering Shortest Path Queries in Planar ...](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... $n - (nr) 1/2$ **partition nodes** • $(nr) 1/2$ **partition boundary nodes** ... Worker Processor – Aux Graph • $(nr) 1/2$ Auxiliary Nodes • n **edges** between these ...

www.jharris.ca/portfolio/acreports/parallelspace.pdf - [Similar pages](#)

[RIOT – The Minimum and Maximum Cut Problems](#)

... Thus, we want to **partition** the **nodes** of the graph into two ... problem by creating a graph that contains a **node** for each data point and an **edge** between each ...

riot.ieor.berkeley.edu/riot/Applications/WeightedMinCut/ - 8k - [Cached](#) - [Similar pages](#)

partition

... The bottleneck graph **partition** problem is to **partition** the **nodes** of a graph into two equally sized sets, so that the maximum **edge weight** in the cut separating ...

riot.ieor.berkeley.edu/~dorit/pub/partition.html - 3k - [Cached](#) - [Similar pages](#)

[[More results from riot.ieor.berkeley.edu](#)]

[Research - Image Segmentation, Normalized Cut](#)

... where entry d_{ii} is the sum of all weights of **edges** originating at **node** i . The ... takes on real values, and the sign is used to **partition** the pixels. ...

www.cs.ucdavis.edu/~jankunm/research/research_ncut.html - 11k - [Cached](#) - [Similar pages](#)

[PS] [An Approximation Algorithm for Maximum](#)

File Format: Adobe PostScript - [View as Text](#)

... It is possible to **partition** $OP \cap T \cap M$ into disjoint subsets S and T so ... $OP \cap T \cap M$ are **node** disjoint; 2. S consists of **node** disjoint 1-**edge** and 2-**edge** paths. ...

www.math.tau.ac.il/~shlomiru/papers/packing.ps - [Similar pages](#)

[Compound Member Index](#)

... **partition**() : Lattice< **NODE**, **EDGE** >; **partition_root** : Lattice< **NODE**, **EDGE** ... Lattice< **NODE**, **EDGE** >; **validate_graph**() : Lattice< **NODE**, **EDGE** >; ... **weight** : balance_data. ...

torch.cs.dal.ca/~panda/pipesort_docs/functions.html - 31k - [Cached](#) - [Similar pages](#)

[PPT] [voor dia serie SNS-Utrecht/'t Gooi](#)

File Format: Microsoft Powerpoint 97 - [View as HTML](#)

... First **node** moving to the growing region is selected by the 1 ... **Partition** Algorithm 4 – IBM90. ... when there are exist two vertices and one **edge** which represents ...

csix.haifa.ac.il/~golumbic/algorclass/mintz2-slides-xfactor.ppt - [Similar pages](#)

VFABC Bucky1 Dataset for FORTE File ...

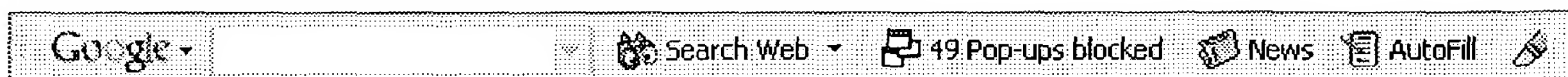
... cada vez que eh dado um assert (**partition**, etc) - 20030201 ... fdt:existsFormerEdge(**node,node**,list(**node**),**edge**,
list(**weight,node,node**)), fdt:notExistsFormerEdge ...

www.cos.ufrj.br/~fcruz/RevisaoTeoria/ExemplosBucky1.dat - 11k - [Cached](#) - [Similar pages](#)

Google

Result Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied?](#) [Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google

Find: [Documents](#)[Citations](#)Searching for **partition and weight and graph**.Restrict to: [Header](#) [Title](#) Order by: [Expected citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

359 documents found. Order: number of citations.

[An Interior-Point Method for Semidefinite Programming - Helmberg, Rendl. \(1996\) \(Correct\) \(131 citations\)](#)

Problem The max-cut problem is the problem of **partitioning** the node set of an edge-weighted undirected problem of **partitioning** the node set of an edge-weighted undirected **graph** into two parts so as to We show that the approach is very efficient for **graph** bisection problems, such as max-cut. Other
orion.math.uwaterloo.ca/~hwoikowi/henry/reports/sdp.ps.gz

One or more of the query terms is very common - only partial results have been returned. Try [Google \(CiteSeer\)](#).

[Normalized Cuts and Image Segmentation - Shi, Malik \(1997\) \(Correct\) \(118 citations\)](#)

an image. We treat image segmentation as a **graph partitioning** problem and propose a novel global in an arbitrary feature space are represented as a **weighted** undirected **graph** $G = (V, E)$ where the nodes of an image. We treat image segmentation as a **graph partitioning** problem and propose a novel global
www.cs.berkeley.edu/~jshi/ftp/Ncut.ps.gz

[Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION - Frieze, Jerrum \(1995\) \(Correct\) \(105 citations\)](#)

guarantees are presented for the problems of (a) **partitioning** the vertices of a **weighted graph** into k the problems of (a) **partitioning** the vertices of a **weighted graph** into k blocks so as to maximise the of (a) **partitioning** the vertices of a **weighted graph** into k blocks so as to maximise the **weight** of
www.math.cmu.edu/~af1p/cuts.ps.gz

[The Markov Chain Monte Carlo Method: An Approach To... - Jerrum, Sinclair \(1996\) \(Correct\) \(63 citations\)](#)

of statistical physics, namely, estimating the **partition** function of a monomer-dimer system. This integration, where the goal is to estimate a **weighted** sum of the form $\sum w(x)$ where w is a approximately counting matchings of all sizes in a **graph**. Section 12.5 then catalogues various other
http.cs.berkeley.edu/~sinclair/mcmc.ps

[Theories for Mutagenicity: A Study in First-Order... - Srinivasan.. \(1995\) \(Correct\) \(56 citations\)](#)

$\log P$: log of the compound's octanol/water **partition** coefficient (hydrophobicity) ffl LUMO : on the back-propagation of errors. Changes in **weight** are calculated by solving a set of differential for which data are most naturally represented by **graphs**, theories constructed with Inductive Logic
ftp.cs.york.ac.uk/pub/ML_GROUP/Papers/ash_aij95.ps.gz

[Approximation Algorithms for Connected Dominating Sets - Guha, Khuller \(1996\) \(Correct\) \(52 citations\)](#)

given an edge **weighted graph** $G = (V, E)$ and a **partition** of $V = V_1 \cup V_2 \cup \dots \cup V_k$ find the of the problem when the vertices have **weights**, and give an algorithm which achieves a MD 20742 Abstract The dominating set problem in **graphs** asks for a minimum size subset of vertices with
ftp.cs.umd.edu/pub/papers/papers/ncstrl.umcp/CS-TR-3660/CS-TR-3660.ps.Z

[On-line Load Balancing - Azar, Broder, Karlin \(1992\) \(Correct\) \(48 citations\)](#)

are only two processors. An easy reduction from **partition**. 1.2 Notation Before plunging into proofs, we requires a different level of service, called the **weight** of the task. Tasks arrive and depart one by one. bandwidth (e.g. low for file transfers, medium for **graphic** applications, high for video) and must be
www.math.tau.ac.il/~azar/bal.ps.gz

[Quick Approximation to Matrices and Applications - Frieze, Kannan \(1999\) \(Correct\) \(39 citations\)](#)

information which implicitly defines the **partition**, enabling its quick construction later. A from certain **graph** problems, such as the maximum **weight** cut problem. Here we have a **graph** $G = (V, E)$ and the fundamental Regularity Lemma of Szemer'edi in **Graph** Theory and the constructive version of Alon,
www.math.cmu.edu/~af1p/matrix.ps.gz

Constraint Satisfaction: The Approximability of.. - Khanna, Sudan, Trevisan (1997) (Correct) (39 citations)
 Our main result is that there exists a finite **partition** of the space of all constraint sets such that satisfies all the constraints while maximizing the **weight** of the assignment (i.e.the number of variables
www.cs.columbia.edu/~luca/pubs/mincsp.ps.Z

On the Hardness of Approximating MAX k-CUT and its Dual - Kann, Khanna, Lagergren.. (1997) (Correct) (33 citations)
 the Max k-Cut problem and its dual, the Min k-**Partition** problem. In the Min k-**Partition** problem, given given a **graph** $G = (V, E)$ and positive edge **weights**, we want to find an edge set of minimum **weight** problem. In the Min k-**Partition** problem, given a **graph** $G = (V, E)$ and positive edge **weights**, we want to
ftp.nada.kth.se/pub/documents/Theory/Viggo-Kann/maxkcut.ps

Multilevel Algorithms for Multi-Constraint Graph Partitioning - Karypis, Kumar (1998) (Correct) (28 citations)
 Multilevel Algorithms for Multi-Constraint **Graph Partitioning** George Karypis and Vipin Kumar
www-users.cs.umn.edu/~karypis/publications/Papers/Postscript/mlevel-mconstraint.ps.gz

The Chaco User's Guide Version 1.0 - Hendrickson, Leland (1994) (Correct) (28 citations)
 Laboratories Albuquerque, NM 87185 Abstract **Graph partitioning** is a fundamental problem in many scientific methods for finding small edge separators in **weighted graphs**. These methods include inertial,
139.179.10.16/prv/ftp/Reports/sandia/Users_Guide.ps.z

VLSI Circuit Partitioning by Cluster-Removal using Iterative.. - Dutt, Deng (1996) (Correct) (27 citations)
 Proc. Physical Design Workshop, 1996 Vlsi Circuit **Partitioning** By Cluster-Removal Using Iterative has all its cells in u's subset. $c(n_i)$ is the **weight** (cost) of the net n_i which is assumed to be
 Lin [1] proposed the well-known KL heuristic for **graph partitioning**. The KL algorithm starts with a
www.eecs.uic.edu/~dutt/.papers/pdw96.ps

A New Approach to the Minimum Cut Problem - David Karger (1996) (Correct) (27 citations)
 and m (possibly **weighted**) edges, we wish to **partition** the vertices into two non-empty sets so as to that finds the minimum cut in an arbitrarily **weighted** undirected **graph** with high probability. The
www.cs.dartmouth.edu/~cliff/papers/Contraction.ps.Z

Dynamic Load-Balancing for Parallel Adaptive Unstructured.. - Walshaw, Cross, Everett (1997) (Correct) (23 citations)
 1997. Abstract A parallel method for dynamic **partitioning** of unstructured meshes is described. The We assume that both vertices and edges are **weighted** (with positive integer values) and that jv_j compared to the static **partitioners**. Key words. **graph-partitioning**, adaptive unstructured meshes,
www.gre.ac.uk/~wc06/papers/ps/WalshawSIAM97.ps.gz

Nonpolyhedral Relaxations of Graph-Bisection Problems - Poljak, Rendl (1992) (Correct) (23 citations)
 $a_{e_1} \leq 2 \leq a_{e_2} \leq 2$ elsewhere. 2) Then the **partition** classes have sizes n_1 and n_2 if and only if
 We consider the problem of decomposing a **weighted graph** G on n nodes into two parts of
 Nonpolyhedral Relaxations Of **Graph-Bisection** Problems Svatopluk Poljak And Franz
fmatbhp1.tu-graz.ac.at/~helmberg/non_poly.ps

A Sublinear Time Approximation Scheme for Clustering in Metric.. - Indyk (Correct) (22 citations)
 is defined as follows: given a metric (X, d) **partition** X into two sets S_1 and S_2 in order to the following k - clustering problem: given a **weighted graph** $G = (X, d)$ on N vertices, where $d(\Delta)$ following k - clustering problem: given a **weighted graph** $G = (X, d)$ on N vertices, where $d(\Delta)$
theory.stanford.edu/~indyk/ptas99.ps

Minimum Cuts in Near-Linear Time - Karger (1998) (Correct) (22 citations)
 and m (possibly **weighted**) edges, we wish to **partition** the vertices into two non-empty sets so as to Given a **graph** with n vertices and m (possibly **weighted**) edges, we wish to **partition** the vertices into for solving the minimum cut problem on undirected **graphs**. We use a "semi-duality" between minimum cuts and
www.uni-paderborn.de/~artur/TEACHING/LITERATUR/NETWORK-ALGORITHMS/Karger-STOC96-lincut.ps.gz

First 20 documents [Next 20](#)

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer - Copyright NEC and IST